UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/081,874	02/21/2002	Nandakumar Gn	Gn 3-3-1-1-1	9904

47386 7590 10/30/2008

RYAN, MASON & LEWIS, LLP
1300 POST ROAD
SUITE 205
FAIRFIELD, CT 06824

EXAMINER	
TRAN, MYLINH T	

ART UNIT	PAPER NUMBER
2179	

MAIL DATE	DELIVERY MODE
10/30/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte NANDAKUMAR GN, SRIRAM GORTI, MOHIT GUPTA, PANKAJ KAKKAR, and CHANDRAMOULEESWARAN SANKARAN

Appeal 2008-1457
Application 10/081,874[1]
Technology Center 2100

Decided: October 29, 2008

Before JEAN R. HOMERE, ST. JOHN COURTENAY III, and THU A. DANG, *Administrative Patent Judges*.

HOMERE, *Administrative Patent Judge*.

DECISION ON APPEAL

I. STATEMENT OF CASE

Appellants appeal under 35 U.S.C. § 134(a) from the Examiner's final rejection of claims 1 through 22. We have jurisdiction under 35 U.S.C. § 6(b). We affirm.

---

[1] Filed on Feb. 21, 2002. The real party in interest is Agere Systems, Inc.

Appellants invented a method and system for generating a graphical interface via a command line interface to facilitate remote access to a software application. (Spec. 2.) As depicted in Figure 5, a web interface generator (500) queries a software developer to specify the name and version of the software application, as well as the machine and directory where the software resides. (*Id.* 8.) The web interface generator (500) further queries the developer to specify the number of option groups, as well as the properties or constraints associated with the option groups for the software application. (*Id.*) Upon receiving all the requested information from the developer, the web interface generator (500) generates a graphical user interface for the software application, and stores it on a remote server (1200) along with a server script to thereby permit clients (110) to remotely execute the software application. (*Id.* 10.)

Independent claim 8 further illustrates the invention. It reads as follows:

8. A method for enabling remote access to one or more software applications having a command line interface, said method comprising the steps of:

querying a user to specify properties of one or more option groups provided by each of said software applications; and

generating a graphical user interface based on said specified properties for each of said software applications, said graphical user interface identifying each of said software applications and allowing one or more clients to remotely access a selected software application.

The Examiner relies on the following prior art as evidence of unpatentability:

Audleman	US 6,806,890 B2	Oct. 19, 2004 (filed Aug. 17, 1999)

The Examiner rejects the claims on appeal as follows:

A. Claims 1 through 22 stand rejected under 35 U.S.C. § 102(b) as being anticipated by Audleman.

FINDINGS OF FACT

The following findings of fact (FF) are supported by a preponderance of the evidence.

Audleman

1. Audleman discloses a system for automatically generating a graphical user interface from a command syntax using an XML document file, an XML schema file, and a text description file. (Col. 2, ll. 18-23.)

2. As depicted in Figure 1, upon receiving XML files representing a command syntax from a command server (110) located on a remote server (106), a client (104) executes a user interface program (114) to process the XML files to thereby identify therefrom resource types, verbs, and keywords. (Col. 3, ll. 11-24.)

3. The user interface program (114) then displays a Wizard that steps a user through the command syntax to thereby generate at least one

command based upon the user's selection from the dialogs. (Col. 3, ll. 24-29.)

4. As shown in Figures 4A-4G, the wizard prompts the user to provide a variety of information pertaining to the transaction to be displayed in order to generate a command for a command processor (108) on the remote server (106) to execute. The requested information includes the name, class number, status, the queue count, the output field of the transaction, as well as the name of the command processor (108) that will execute the generated command. (Col. 8, l. 36–col. 9, l. 9.)

5. Upon receiving the generated command from the user interface program (114), the command server (110) forwards it to the command processor (108) to remotely execute a desired software application on behalf of the user. (Col. 3, ll. 31-38.)

PRINCIPLES OF LAW

ANTICIPATION

In rejecting claims under 35 U.S.C. § 102, "[a] single prior art reference that discloses, either expressly or inherently, each limitation of a claim invalidates that claim by anticipation." *Perricone v. Medicis Pharmaceutical Corp.*, 432 F.3d 1368, 1375 (Fed. Cir. 2005), citing *Minn. Mining & Mfg. Co. v. Johnson & Johnson Orthopaedics, Inc.*, 976 F.2d 1559, 1565 (Fed. Cir. 1992). "Anticipation of a patent claim requires a finding that the claim at issue 'reads on' a prior art reference." *Atlas Powder*

Co. v. IRECO, Inc., 190 F.3d 1342, 1346 (Fed Cir. 1999) ("In other words, if granting patent protection on the disputed claim would allow the patentee to exclude the public from practicing the prior art, then that claim is anticipated, regardless of whether it also covers subject matter not in the prior art.") (internal citations omitted).

The *claims* measure the invention. *See SRI Int'l v. Matsushita Elec. Corp., of America*, 775 F.2d 1107, 1121 (Fed. Cir. 1985) (en banc). "[T]he PTO gives claims their 'broadest reasonable interpretation.'" *In re Bigio*, 381 F.3d 1320, 1324 (Fed. Cir. 2004) (quoting *In re Hyatt*, 211 F.3d 1367, 1372 (Fed. Cir. 2000)). "Moreover, limitations are not to be read into the claims from the specification." *In re Van Geuns*, 988 F.2d 1181, 1184 (Fed. Cir. 1993) (citing *In re Zletz*, 893 F.2d 319, 321 (Fed. Cir. 1989)).

ANALYSIS

Independent claim 8 recites in relevant part querying a user to specify the properties of an option group provided by a software application. (Claims Appendix.)

Appellants argue that Audleman does not teach these limitations. (App. Br. 6.) Particularly, Appellants argue that Audleman does not teach option groups as defined in the Specification. (*Id.*) Appellants further argue that Audleman does not teach specifying properties for the option groups. (*Id.*) Therefore, Appellants submit that Audleman does not anticipate the cited claim.

In response, the Examiner avers that Audleman's disclosure of identifying resource types, verbs and keywords in the command syntax of processed XML files teach the option groups and the properties associated therewith. (Ans. 8-9.)

Therefore, the pivotal issue before us is whether Audleman's disclosure teaches querying a user to specify properties of an option group provided by a software application, as recited in independent claim 8. We answer this inquiry in the affirmative.

We begin by considering the scope and meaning of the afore-cited limitations, which must be given their broadest reasonable interpretation consistent with Appellants' disclosure, as explained in *In re Morris*, 127 F.3d 1048, 1054 (Fed. Cir. 1997):

> [T]he PTO applies to the verbiage of the proposed claims the broadest reasonable meaning of the words in their ordinary usage as they would be understood by one of ordinary skill in the art, taking into account whatever enlightenment by way of definitions or otherwise that may be afforded by the written description contained in the applicant's specification.

Id. at 1054. *See also In re Zletz*, 893 F.2d at 321 (Fed. Cir. 1989) (stating that "claims must be interpreted as broadly as their terms reasonably allow." Appellants' Specification states the following:

> The developer is then queried during step 540 to *specify the properties of each option group, i.e., for the constraints associated with a given option group*, such as whether the various options within an option group can be used together and any input file requirements. [Emphasis added.]

(Spec. 8, ll. 15-18.)

> The software applications have the following general syntax: Tool_name [option 1] [option 2] . . . <filename>] where *each of these options further can be of one of the following types {exactly one parameter; one or more than one; none or more and with or without an input file}*. In this manner, the developer 210 or administrator can establish groups and subgroups of parameters with similar properties. [Emphasis added.]

(Spec. 8, l. 28- spec. 9, l. 4.)

Our reviewing court further states, "the 'ordinary meaning' of a claim term is its meaning to the ordinary artisan after reading the entire patent." *Phillips v. AWH Corp.,* 415 F.3d 1303, 1321 (Fed. Cir. 2005.)

Upon reviewing Appellants' Specification, we find that an 'option' can be 'no parameter', or 'at least one parameter with or without an input file'. Similarly, we find an option group to be a set of options with similar properties or constraints specified and grouped by a developer. Consequently, consistently with Appellants' Specification, we broadly but reasonably construe an option group as either an empty set, or a set of at least one parameter having a specified constraint. We note that when the option group is an empty set, the claim is indefinite. Alternatively, when the option group contains at least one parameter having a constraint, we find that the claimed limitation in question merely requires querying a user to specify a constraint for a parameter provided by a software application. *See In re Bigio*, 381 F.3d at 1324.

Claims 1-10, 12-20, and 22

As set forth in the Findings of Fact section, Audleman discloses in response to a syntax command issued by a client, a server computer provides the client with an XML file containing certain resource types, verbs, and keywords associated with the client command. (FF. 1-2.) Further, Audleman discloses stepping the client through various Wizard dialogs to select desired parameters in order to create corresponding commands. (FF. 3-4.) We find that the resource types, verbs and keywords returned to the client user in response to an issued command syntax are particularly associated with a desired software program. In other words, the user's selection of different command syntaxes will result in returning to the user different resource types, verbs and keywords associated therewith. Therefore, by dispatching a particular command syntax, the user indirectly specifies certain parameters for the desired software. Therefore, we agree with the Examiner's initial showing, in the limited situation where the option group is not an empty set, that Audleman's resource types, verbs and keywords are parameters specified by a user to generate an interface for a particular software program. Accordingly, we find that the Examiner has set forth a sufficient initial showing of anticipation of independent claim 8. We also find that Appellants have not shown that the Examiner erred in finding that Audleman anticipates independent claim 8.We therefore sustain the rejection of claim 8.

Appellants do not provide separate arguments with respect to the rejection of claims 1 through 7, 9, 10, 12 through 20, and 22. Consequently, these claims fall together with independent claim 8. 37 C.F.R. § 41.37(c)(1)(vii).

Claims 11 and 21

Appellants argue that Audleman does not teach a remote server script that provides any necessary input to a remote server to initiate the execution of a selected software application thereon. (App. Br. 6.) We do not agree. As set forth in the Findings of Facts section, Audleman discloses a user interface that supplies a generated command to a remote server in order to permit the user to remotely execute a corresponding software application thereon. We find that the server's transfer of a desired software application in response to the user's issued command is done by executing a remote server script. It follows that Appellants have not shown that the Examiner erred in finding that Audleman anticipates claims 11 and 21.

CONCLUSIONS OF LAW

Appellants have not shown that the Examiner erred in finding that claims 1 through 22 are anticipated under 35 U.S.C. § 102.

DECISION

We affirm the Examiner's decision rejecting claims 1 through 22.

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a)(1)(iv).

AFFIRMED

rwk

RYAN, MASON & LEWIS, LLP
1300 POST ROAD
SUITE 205
FAIRFIELD CT 06824